Universitat de Lleida

TREBALL FINAL DE MÀSTER

ESCOLA
POLITÈCNICA SUPERIOR
UNIVERSITAT DE LLEIDA
INSPIRING THE FUTURE

Estudiant: **Meritxell Jordana Gavieiro**

Titulació: Màster en Enginyeria Informàtica

Títol de Treball Final de Màster: Xarxes vehiculars: Implementació d'un protocol criptogràfic basat en la Identitat

Director/a: Josep M. Miret Biosca

Presentació

Mes: Juliol

2018

Xarxes vehiculars. Implementació d'un protocol criptogràfic basat en la identitat

Autor: Meritxell Jordana Gavieiro
Director: Josep M. Miret Biosca

Universitat de Lleida
Escola Politècnica Superior
Màster en Enginyeria Informàtica

Treball Final de Màster

10 de juliol de 2018

Índex

Índex de figures

Capítol 1

Introducció

Estem en una època en la que cada vegada hi ha més vehicles en circulació, així doncs, cada vegada hi ha un risc més elevat de possibles incidents on aquests vehicles estiguin implicats. Com a incidents podem entendre possibles accidents causats, per exemple, per xocs entre ells, possibles avaries que dificultin la circulació de la resta o fins hi tot possibles embussos.

Amb una quantitat tan gran de vehicles en circulació, és normal la presència d'individus que vulguin aprofitar-se de possibles vulnerabilitats que puguin tenir els sistemes de seguretat d'aquests vehicles per poder aprofitar-se d'algunes situacions i, fins i tot, poder causar alguns incidents. Per evitar aquestes possibles intrusions als sistemes de seguretat dels vehicles, necessitem conèixer i estudiar nous mecanismes de seguretat que ens permetin cada vegada garantir un nivell de seguretat més elevat.

En aquest treball estudiarem mecanismes de seguretat que utilitzen la criptografia basada en la identitat. He decidit treballar amb la criptografia basada en la identitat ja que vaig introduir-me en ella durant el treball final de grau i volia continuar per explorar més camps on pot ser aplicada i útil. En el treball, a part del treball teòric on s'explicaran els conceptes fonamentals de la criptografia basada en la identitat i com és aplicada en xarxes vehiculars, també es farà referència a un parell de papers que han estat estudiats en profunditat i en els quals se'ns ofereixen nous mecanismes en diferents camps de les xarxes vehiculars. Aquestes lectures s'han trobat molt útils de cara a la implementació final que es farà, on es mostrarà un esquema en el que es combina la utilització de la signatura basada en la identitat i un nou

tipus de signatura basada en la identitat que està composta per dues fases; una offline i una altra online. Aquest esquema a part de servir-nos de guia per entendre les comunicacions dins d'aquestes xarxes, també ens mostrarà dos tipus diferents d'autenticacions que poden extendre la seva utilització a altres camps diferents a les xarxes vehiculars.

Capítol 2

Introducció a les VANETs

2.1 Qué són les VANETs?

Una xarxa vehicular, coneguda en sigles angleses com a VANET (Vehicular Ad-hoc NETwork) és, com el seu nom indica, una xarxa **ad-hoc** on els seus nodes són vehicles. Una xarxa ad-hoc és un tipus de xarxa en la que no hi ha un node central, sinó que els dispositius estan en igualtat de condicions. A més a més, aquests dispositius no depenen de cap infrastructura.

També cal fer un incís en que una VANET és un xarxa del tipus **MANET** (Mobile Ad-hoc NETwork). Les xarxes MANET són xarxes de nodes mòbils interconectats amb enllaços de comunicació inalàmbrica multisalts, que a diferència de les xarxes inalàmbriques convencionals, les xarxes Ad-hoc no tenen una intrastructura fixa de xarxa o administració central, on la topologia d'aquestes xarxes canvia dinàmicament amb nodes mòbils entrant i sortint de la xarxa.

L'estudi de les xarxes vehiculars és necessari per molts motius, entre els quals trobem que:

- La seguretat vial es un problema vital. Existeixen múltiples ocasions en les que les comunicacions entre vehicles ajudarien a prevenir possibles accidents, així com evitar embussos en la carretera.

- Cada vegada hi ha més vehicles en moviment i el risc de possibles accidents va en augment.

- Un percentatge molt alt d'accidents són provocats pel factor humà, així doncs, una part dels accidents seria "evitable".
- L'ajuda de la tecnologia podria prevenir i evitar alguns d'aquests accidents alertant al conductor dels riscos de la carretera.
- Les VANETs ajudarien a evitar les retencions: estalviant temps, diners, reduint la contaminació del medi ambient i ajudant les reserves de petroli.

2.1.1 Descripció dels components de la xarxa

Els nodes que formen part de la xarxa poden dividir-se en dos grans grups:

1. **Nodes mòbils:** Com ja s'ha explicat en l'anterior definició de VANET, els vehicles són la part fonamental de la xarxa i aquests, estan en continu moviment. Aquests vehicles venen equipats amb un dispositiu electrònic anomenat **OBU** (On Board Unit) per poder comunicar-se tant amb els altres vehicles com amb els nodes estàtics que explicarem a continuació. Aquests nodes mòbils tenen la capacitat d'enviar, rebre i transmetre missatges entre ells o recolzar-se en els nodes estàtics. Aquests tipus de comunicacions també es detallaran més endavant en la secció. També és necessari comentar que els OBUs envien missatges broadcast [1] periòdics amb informació sobre la seva posició, temps, direcció, velocitat, etc
2. **Nodes estàtics:** Són elements fixos que estan colocats al llarg de les carretes i que s'anomenen **RSU** (Road-Side Unit), els quals tenen la funció d'enviar, rebre i transmetre paquets per augmentar el rang de cobertura de la xarxa, podent també oferir accés a Internet. Els RSU envien missatges broadcast relacionats amb el tràfic, especialment útils en cas de retencions o accidents.

A part dels nodes que formen part de la xarxa de manera més visible, també tenim altres components de vital importància:

- Es compta amb la presència d'una **TA** (Trusted Authority) que anomenarem més endavant **RTA** (Regional Trusted Authority) , que és la

[1] És la difusió d'un missatge a **tots** els components d'una xarxa

unitat central en la que es pot confiar i la qual ens proveirà els registres i certificacions dels RSUs i els OBUs al unir-se a la xarxa.

- També es disposa d'una **CRL** (Certificate Revokation List). S'utilitzarà en dos possibles casos:

  - En cas de que arribi un missatge d'una entitat desconeguda, el vehicle ha de consultar aquesta llista per evitar comunicar-se amb vehicles revocats.
  - Després de verificar la signatura de l'emisor per a verificar la validesa del missatge rebut

  L'únic inconvenient que els genera la presència d'aquesta CRL és el temps de consulta dins d'aquesta llista. Aquest temps és massa alt i podueix un delay que s'ha de tractar de reduir-lo.

Un cop definits els principals components que faran possible la comunicació, definirem els tres tipus de comunicacions que es poden produir entre aquests components:

1. **V2V:** Aquestes sigles angleses es llegeixen com Vehicle-to-Vehicle, així doncs, aquest tipus de comunicació és la que es produirà entre els vehicles de la xarxa.

2. **V2I:** Aquestes sigles angleses es llegeixen com Vehicle-to-Infrastructure, així doncs, aquest tipus de comunicació és la que es produria entre les unitats a bord dels cotxes (OBU) i les unitats que es troben al costat de les carreteres (RSU) i amb els RTA.

3. **I2I**: Aquestes sigles angleses es llegeixen com Intrastructe-to-Infrastructure, així doncs, podem deduïr que les unitats RSU també es comunicaran entre elles per intercanviar informació important, així com també amb el RTA.

2.2 Problemes que poden presentar les VANETs

Entre els principals reptes que presenten les xarxes vehiculars podem destacar els següents problemes:

- La manca d'una infrastructura central que estigui a càrrec de la sincronització i la coordinació de les transmissions fa que una de les tasques més complexes sigui la gestió del canal (de la xarxa inalàmbrica) per aconseguir un ús eficient de l'ample de banda.

- L'alta mobilitat dels nodes, els requisits per a l'escalabilitat i la gran varietat de condicions ambientals són tres dels reptes més importants de les xarxes descentralitzades i autoorganitzades. Un problema en particular al que s'ha de fer front prové de les altes velocitats dels vehicles en alguns trams, com per exemple en les autopistes (120 km/h en autopistes espanyoles, per exemple).

- Els requisits de seguretat i privacitat en VANETs han d'equilibrar-se. Per una banda tenim que els receptors volen assegurar-se que poden confiar en la procedència de la informació (emisor), mentre que per una altra banda tenim que això podria estar en desacord amb els requisits de privacitat de l'emisor.

- La necessitat d'estandarització de les comunicacions en les xarxes vehiculars hauria de permetre flexibilitat, ja que aquestes xarxes han de permetre operar amb moltes marques diferents d'equips i fabricants de vehicles.

- Les comunicacions en temps real són una condició necessària perque no pot existir retràs en la transmissió d'informació relacionada amb la seguretat. Això implica que les comunicacions VANET requereixen processament i intercanvi d'informació ràpids.

- Les comunicacions per a l'intercanvi d'informació són basades en connexions node a node. Tenir una xarxa distribuïda implica que els nodes han de retransmetre a altres nodes missatges per prendre decisions sobre la ruta a elegir i també que qualsevol node pot actuar com a un host que demana informació o com un router que distribueix informació, depenent de les circumstàncies.

2.3 Atacs a VANETs

Com ja s'ha fet esment a l'inici d'aquest treball, les xarxes VANET són molt importants per poder realitzar comunicacions entre vehicles però, com

a tota comunicació, sempre hi pot haver intrusions que puguin posar en perill o alterar aquestes comunicacions. En aquesta secció el que s'intentarà és fer esment dels atacs més comuns que pateixen les comunicacions que es produeixen en les VANETs. Aquests atacs es classifiquen segons què posen en perill de la comunicació, i de cada subgrup s'explicaran les més importants i més utilitzades:

1. **Disponibilitat**. És un dels factors més importants en VANETs. Garanteix que una xarxa és funcional i la informació important està sempre disponible. Alguns dels atacs més importants són els següents:

   (a) **DoS (Deny of Service)**. Aquest atac consisteix en inundar de tràfic un sistema o una xarxa fins a que no sigui capaç de donar servei a usuaris legítims. Al crear tant tràfic poden passar dues coses: que les màquines responsables de contestar peticions no donin més de si o que l'ample de banda de la xarxa no pugui processar tantes dades. Al no poder seguir donant un servei, es diu que aquest ha estat denegat. A més a més, també comptem amb la presència d'un atac DoS distribuït (DDoS), el qual és més difícil de gestionar que un atac centralitzat donat que el tràfic es genera desde varis punts. Jamming, Greedy behavior i Blackhole attack són exemples d'atacs DoS.

   (b) **Jamming**. Aquests tipus d'atacs desactiven o saturen els recursos del sistema. Per exemple, un atacant pot consumir tota la memòria o espai en disc disponible, així com enviar tant tràfic a la xarxa que ningú més pugui utilitzar-la. Aquí, l'atacant satura el sistema amb missatges que necessiten establir connexió, no obstant, enlloc de donar la direcció IP real de l'emisor, el missatge conté falses direccions IP. El sistema respon el missatge però no obté resposta i això fa que s'acumuli buffers amb informació de les connexions obertes, no deixant lloc a les connexions legítimes.

   (c) **Greedy behavior**. És un atac en la funcionalitat de la capa MAC segons l'arquitectura del model OSI. El node «greedy» (avariciós) no respecta el mètode d'accés al canal i sempre intenta connectarse. El principal motiu és evitar que els demés nodes puguin utilitzar els serveis. El greedy behavior provoca sobrecàrrega i problemes de col·lisió en el medi de transmissió, el qual provoca retards en els serveis per a usuaris autoritzats.

(d) **Blackhole attack**. És un atac convencional en contra de la disponibilitat en les xarxes ad-hoc, així doncs també per VANETs. El node maliciós rep paquets de la xarxa, però refusa participar en les operacions d'enrutament. Això altera les taules d'enrutament i evita l'arribada de dades importants al receptor principalment perque el node el node maliciós sempre declara formar part de la xarxa i ser capaç de participar, el qual no és el cas a la pràctica.

2. **Autenticitat i identificació**. És un dels reptes més importants en VANETs (seguretat). Totes les estacions s'han d'autenticar abans d'accedir als serveis i, per tant, qualsevol atac o violació comporta una exposició de tota la xarxa. Alguns dels atacs més importants són els següents:

   (a) **Sybil Atack**. En aquest atac, l'atacant pot contaminar un sistema distribuït creant un gran número d'identitats que aparenten ser independents i utilitzar-les per obtenir una influència desproporcionada, alterant rutes o modificant contingut emmagatzemat de forma redundant. D'aquesta manera, certs nodes legítims poden patir una usurpació de la identitat a l'estar connectats als de l'atacant. La vulnerabilitat del sistema depen de la facilitat per crear noves identitats i la importància de la cadena de confiança, que pot fer que totes les identitats siguin tractades per igual.

   (b) **GPS spoofing**. Un atac GPS spoofing el que fa és distreure l'atenció d'un receptor GPS per així poder suplantar la senyal original amb la senyal fraudulenta d'un tercer, de tal forma que el receptor no podrà descobrir que aquesta senyal ha canviat de procedència. Això s'aconsegueix construint una senyal fraudulenta i, poc a poc, es va augmentant l'energia de transmissió de l'ona. Quan la senyal fraudulenta és més forta que l'original en un o més satèl·lits GPS, l'ona s'acopla sobre la original i aquesta serà absorbida, quedant únicament la senyal modificada.

   (c) **Tunneling**. L'atac consisteix en connectar dues parts distants de la xarxa vehicular utilitzant la mateixa xarxa per establir una connexió privada (el que es coneix com a túnel). Com a conseqüència, les víctimes de les dues parts distants de la xarxa poden comunicar-se com si fossin veïns.

(d) **Key/certificate replication**. L'atac consisteix en la utilització de claus duplicades i/o certificats utilitzats per demostrar la identitat i per crear ambigüetat que el que fa és dificultar a les autoritats la identificació del vehicle, especialment en el cas d'alguna disputa.

3. **Confidencialitat**. Consisteix en assegurar que les dades només són llegides per parts autoritzades. Pot afectar a la privacitat dels individus perquè pot comportar la compartició de dades importants com rutes i informació privada del conductor. Alguns dels atacs més importants són els següents:

   (a) **Eavesdropping**. En les xarxes inalàmbriques com les VANETs, escoltar el medi és un atac fàcil de realitzar. A més a més, és passiu i la víctima no n'és conscient. L'atac és contra la confidencialitat però no és un atac inminent en la xarxa. A través d'aquest atac, algunes dades molt importants són recollides, com per exemple les dades de localització que poden ser utilitzades per rastrejar vehicles.

   (b) **Traffic analysis attack**. En VANETs, l'atac basat en l'anàlisi de tràfic és una amenaça passiva seriosa contra la confidencialitat i la privacitat dels usuaris. L'atacant analitza dades recollides després d'una fase d'escolta a la xarxa, i tracta d'extreure el màxim d'informació útil per als seus propòsits.

4. **Integritat i dades fiables**. Consisteix en assegurar que les dades no han estat alterades durant les comunicacions V2V i V2I (possible manipulació en els sensors del vehicle). Alguns dels atacs més importants són els següents:

   (a) **Masquerading**. En aquest atac, l'atacant s'oculta utilitzant una identitat vàlida (anomenada màscara), i intenta crear un "black-hole" o produir falsos missatges que semblen venir d'un node autèntic. Per exemple, reduir la velocitat d'un vehicle o requerir un canvi de carril. Un node maliciós intenta fer-se passar per un vehicle d'emergència i amb això enganyar els altres vehicles.

   (b) **Replay attack**. Aquest és un atac molt clàssic, el qual consisteix en fer broadcast d'un missatge ja enviat per aprofitar-se del

missatge en el moment d'entregar-lo. Així, l'atacant injecta a la xarxa paquets prèviament rebuts. Amb això, per exemple, es poden manipular les localitzacions i les taules d'enrutament dels nodes. Al contrari d'altres atacs, l'atac per replay pot executar-se per usuaris no legítims.

(c) **Message supression/Fabrication/Alteration**. Com el nom indica, aquest atac contra la integritat consisteix en modificar o alterar les dades existents. Pot passar modificant una part específica del missatge que ha de ser enviat. Per exemple, l'atacant falsifica les dades rebudes indicant que la ruta no està congestionada quan el missatge original així ho indicava, i d'aquesta manera enganya a la resta d'usuaris. També podria eliminar part del missatge per als seus propòsits o crear-ne de nous.

5. **No-repudi i responsabilitat**. És l'habilitat de verificar que l'emisor i el receptor són realment qui diuen ser en l'emissió i la recepció de missatges. L'atac més important és el següent:

   (a) **Loss of event traceability**. Els atacs de no repudi consisteixen en prendre mesures, permetent posteriorment que un atacant negui haver realitzat una o més accions. Aquest tipus d'atac es basa essencialment en l'eliminació dels rastres d'accions i en la creació de confusions per part de l'entitat atacant. Per exemple, un vehicle no ha de ser capaç de negar-se a enviar un warning.

2.4 Estat de l'art actual

En l'última dècada, les xarxes vehiculars han començat a despertar un gran interès tant a nivell automobilístic com a nivell de investigació, perque com ja s'ha esmentat, la quantitat de vehicles en circulació va augmentant de manera força ràpida, el que implica que el nivell d'incidents també augmenta, cosa que es vol estudiar per poder reduir aquest nombre.

Actualment es podria dir que està en fase d'investigació, fet que augmenta la quantitat d'oportunitats. També és cert que grans companyies multinacionals com Cisco, Google, Microsoft, Toyota, Renault (podem veure que hi ha tant companyies automobilístiques com companyies tecnològiques, ja que

ambdós poden treure profit de les xarxes vehiculars i el seu desenvolupament) lideren les principals iniciatives.

Com ja s'ha comentat en un dels possibles problemes que comporten les VANETs, hi ha un procés d'estandarització que és necessari que es produeixi pel sol fet de que hi ha moltes marques que fabriquen components electrònics i molts fabricants de vehicles, i hi hauria d'haver flexibilitat per a que hi hagués compatibilitat absoluta. Aquest procés d'estandarització està generant problemes tant econòmics, legals com institucionals que l'únic que estan fent és endarrerir-ho.

Un cop estigui solucionat aquest problema d'estandarització, el procés de llançament hauria de començar, fent que fós tot completament funcional en pocs anys.

Capítol 3

Criptografia basada en la identitat

En aquest capítol s'intentaran deixar ben definits els principals conceptes relacionats amb la criptografia. En primer lloc es farà un repàs a les corbes el·líptiques i les seves propietats, en segon lloc els pairings i les propietats i per últim es parlarà de la criptografia basada en la identitat.

3.1 Corbes el·líptiques: Introducció

Una corba el·líptica sobre un cos $\mathbb{K}$ és una corba algebraica sense punts singulars que ve donada per una equació del tipus

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6, \quad a_i \in \mathbb{K} \tag{3.1}$$

denominada equació general de Weierstrass. Si la característica de $\mathbb{K}$ és diferent de 2 i 3, l'equació de la corba es pot expressar com

$$y^2 = x^3 + ax + b, \quad a, b \in \mathbb{K} \tag{3.2}$$

anomenada equació reduïda de Weierstrass. Per a que això es compleixi, es necessita que el discriminant del polinomi cúbic en x no sigui nul, per tant, $\triangle = 4a^3 + 27b^2 \neq 0$. Si es disposa d'aquesta característica, la corba no té singularitats (punts singulars). En la Figura 3.1 es pot veure un exemple de corba el·líptica sobre $\mathbb{R}$ que té com a equació: $y^2 = x^3 - 13x - 12$.

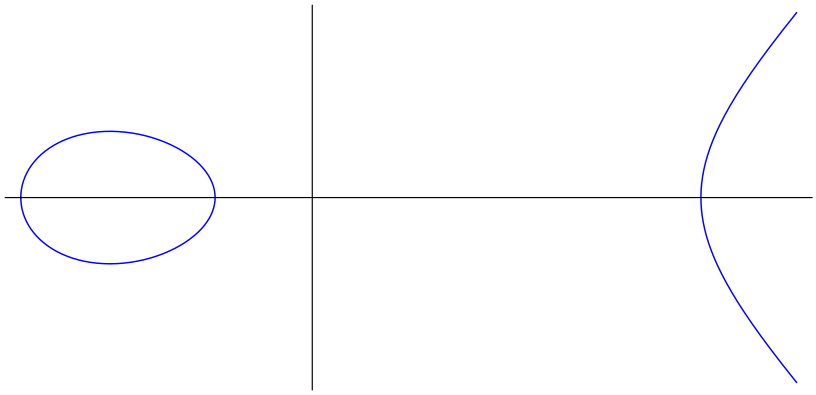

Figura 3.1: Gràfica de la corba $y^2 = x^3 - 13x - 12$ definida sobre $\mathbb{R}$

Si $E/\mathbb{K}$ és una corba el·líptica sobre un cos $\mathbb{K}$, denotarem per $E(\mathbb{K})$ el conjunt de punts $P = (x, y) \in \mathbb{K} \times \mathbb{K}$ que satisfan l'equació de la corba juntament amb el punt de l'infinit de la corba ($\mathcal{O}$). Expressat de manera formal tindríem que :

$$E(\mathbb{K}) = \left\{(x, y) \in \mathbb{K} \times \mathbb{K} \mid y = x^3 + ax + b,\ a, b \in \mathbb{K}\right\} \cup \{\mathcal{O}\}\,. \tag{3.3}$$

Per facilitar els càlculs prenem com a punt base el punt de l'infinit de la corba, $\mathcal{O} = (0 : 1 : 0)$ en coordenades projectives.

3.1.1 Suma de punts en una corba el·líptica

Al conjunt de punts d'una corba $E/\mathbb{K}$ podem definir-hi una operació suma, que podem expressar de forma gràfica o de forma algebraica, tal i com es mostrarà a continuació.

Mètode de la corda i la tangent

Aquest mètode consisteix en traçar una recta que uneixi dos punts P, Q que són els que volem sumar. Aquesta recta talla un tercer punt de la corba, el qual anomenarem R. Sabem que hi haurà aquest tercer punt de tall perquè si tenim una corba de grau 3 i una recta de grau 1, aleshores hi haurà 3 punts d'intersecció entre ambdues. Si P i Q són el mateix punt, aleshores s'agafaria com a recta la tangent a la corba en el punt.

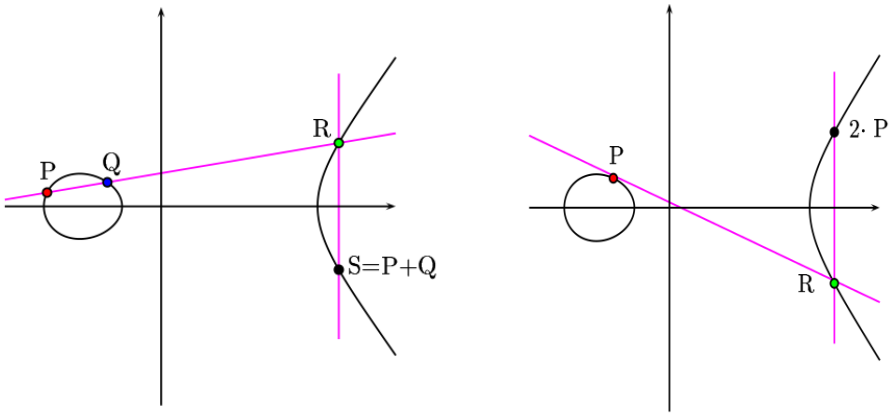

Figura 3.2: P+Q i P+P amb el mètode la corda i la tangent

El punt $P + Q$ (o el punt $2 \cdot P = P + P$) és el punt d'intersecció de la corba amb la recta que passa per R i $\mathcal{O}$, és a dir, la recta que passa per R i es paral·lela a l'eix d'ordenades. En la Figura 3.2 es pot observar com a partir dels punts P i Q de la corba i el punt R també de la corba, aconseguim generar el punt S que serà el punt suma de $P+Q$. També es pot veure, de manera similar, com funciona el doblat del punt P també utilitzant el mateix mètode.

Amb aquesta operació, $(E/\mathbb{K}, +)$ té estructura de grup abelià amb el punt de l'infinit $\mathcal{O}$, com a punt neutre.

3.1.2 Càlcul algebraic de la suma de dos punts

Les expressions de les coordenades del punt $P+Q = (x_3, y_3)$, si $P+Q \neq \mathcal{O}$, en termes de $P = (x_1, y_1)$ i $Q = (x_2, y_2)$, venen donades per:

$$\begin{aligned} x_3 &= \lambda^2 - x_1 - x_2, \\ y_3 &= (x_1 - x_3)\lambda - y_1, \end{aligned}$$

on λ és la pendent de la recta que passa per P i Q (recta tangent a P si $P = Q$) i el seu valor és:

- $\lambda = \frac{(y_1 - y_2)}{(x_1 - x_2)}$ si $x_1 \neq x_2$ o

- $\lambda = \frac{(3x_1^2+a)}{(2y_1)}$ si $x_1 = x_2$ i $y_1 \neq 0$

Substituint el valor de λ en les equacions anteriors, obtindrem les coordenades del punt suma.

3.1.3 Múltiples d'un punt

Utilitzant l'operació suma definida anteriorment, es pot definir el producte de $n \in \mathbb{Z}$ i un punt $P \in E/\mathbb{K}$ de la següent forma:

$$n \cdot P = \begin{cases} \overbrace{P + P + P + ... + P}^{n \ vegades} & si \ \ n > 0, \\ \overbrace{(-P) + ... + (-P)}^{|n| \ vegades} & si \ \ n < 0, \\ \mathcal{O} & si \ \ n = 0. \end{cases}$$

Quan tenim un enter n molt gran, el mètode de calcular $n \cdot P$ sumant n vegades P és molt ineficient, ja que té cost $O(n)$, que és molt més gran que el cost que tindria per exemple un algoritme com el **camperol rus** que tindria un cost de $O(\log_2 n)$.

3.1.4 Corbes el·líptiques sobre cossos finits

Una corba el·líptica sobre un cos finit $\mathbb{F}_q$, anomenada a partir d'ara $E/\mathbb{F}_q$, on $q = p^m$ i p és un nombre primer, ve definida per una equació de la forma:

$$y^2 = x^3 + ax + b \tag{3.4}$$

on a, b són elements de $\mathbb{F}_q$ i $4a^3 + 27b^2 \not\equiv 0$. Aquestes corbes proporcionen grups finits de gran interès criptogràfic degut a la dificultat que té el problema del logaritme discret plantejat sobre ells. Normalment es consideren com a cos base $\mathbb{F}_p$, amb una p gran, o $\mathbb{F}_{2^m}$, amb m gran.

Totes les propietats vistes anteriorment sobre cossos, també són aplicables a les corbes el·lítiques sobre cossos $\mathbb{F}_q$, a més a més d'una sèrie de propietats i característiques pròpies.

Algoritmes per calcular el cardinal

El cardinal d'una corba el·líptica $E/\mathbb{F}_q$, que denotarem per $\#E(\mathbb{F}_q)$ és el nombre de punts que conté la corba amb coordenades a $\mathbb{F}_q$, més el punt a l'infinit $\mathcal{O}$. Per tant, com a mínim el nombre de punts de la corba és $N \geq 1$. Prenem ara, $x \in \mathbb{F}_q$ (x pot prendre per valor q diferents valors), si $\exists y \in \mathbb{F}_q$ tal que $y^2 = x^3 + ax + b$, llavors $-y$ també compleix l'equació i, per tant, podem dir que $N \leq 1 + 2q$.

Per acotar més els càlculs del raonament anterior, s'utilitza el teorema de Hasse:

Teorema 1. *(de Hasse). Sigui E una corba el·líptica definida sobre un cos finit $\mathbb{F}_q$ i sigui $N = \#E(\mathbb{F}_q)$, llavors:*

$$q + 1 - 2\sqrt{q} \leq N \leq q + 1 + 2\sqrt{q}$$

és a dir, $N = q + 1 - t$ amb $|t| < 2\sqrt{q}$. A l'enter t se l'anomena traça de l'endomorfisme de Frobenius de $E/\mathbb{F}_q$.
Es diu que $E/\mathbb{F}_q$ és supersingular si $t \equiv 0 \pmod{q}$. En cas contrari es diu que és ordinària.

Teorema 2. *(de Cassels). Sigui E una corba el·líptica sobre $\mathbb{F}_q$.*

$$E(\mathbb{F}_q) \text{ és isomorf a } \begin{cases} \mathbb{Z}_m & \text{on } m = \#E(\mathbb{F}_q) \\ & \text{o} \\ \mathbb{Z}_{m_1} \times \mathbb{Z}_{m_2} & \text{on } \begin{cases} m_1 \cdot m_2 = m = \#E(\mathbb{F}_q) \\ m_2 | m_1 \\ m_2 | (q-1) \end{cases} \end{cases}$$

El càlcul del cardinal d'una corba el·lípica és una tasca computacionalment molt costosa, és per això que el mètode del recompte exhaustiu no s'utilitza, sinó que s'utilitzen d'altres algorismes com el SEA(Schoof–Elkies–Atkin) de Schoof [16] o el de Shanks-Mestre(o *baby-step giant-step)* [7].
L'algorisme de Shanks-Mestre té complexitat $O(q^{\frac{1}{4}+\varepsilon})$, on ε és una constant positiva que es pot fer arbitràriament petita. En aplicacions criptogràfiques s'utilitza un mètode amb complexitat $O(log^8 q)$. Aquesta millora es deu al treball de Schoof. La idea és computar l'ordre del grup mòdul primers petits i aleshores utlitzar el *Chinese Remainder Theorem (* Teorema xinès del residu) per obtenir l'ordre exacte. En [3] es pot trobar una àmplia explicació sobre el funcionament de l'algoritme així com les demostracions necessàries per veure que l'algoritme funciona correctament. Aquí no s'entrarà en més detalls ja que no és la veritable finalitat del treball.

3.2 Criptografia amb pairings: Introducció

En aquesta secció introduirem els conceptes principals de pairings. Per a més detall vegi's [1, 8, 9, 14].

3.2.1 Pairings

Definició. Considerem dos grups additius $(G_1, +)$ i $(G_2, +)$ amb neutre 0 i un grup multiplicatiu $(G_T, \cdot)$ amb neutre 1. Un pairing o aparellament és una aplicació

$$e : G_1 \times G_2 \to G_T$$

que satisfà les següents propietats:

1. (Bilinealitat).
   $\forall S_1, S_2 \in G_1, \forall T \in G_2, \forall a_1, a_2 \in \mathbb{Z}\ , e(a_1 S_1 + a_2 S_2, T) = e(S_1, T)^{a_1} e(S_2, T)^{a_2}$
   $\forall S \in G_1, \forall T_1, T_2 \in G_2, \forall b_1, b_2 \in \mathbb{Z}\ , e(S, b_1 T_1 + b_2 T_2) = e(S, T_1)^{b_1} e(S, T_2)^{b_2}$

2. (No degeneració). $\forall S \in G_1 \ \ S \neq 0$, existeix $T \in G_2$ tal que $e(S, T) \neq 1$.
   De la mateixa forma també tenim que:
   $\forall T \in G_2 \ \ T \neq 0$, existeix $S \in G_1$ tal que $e(S, T) \neq 1$.
   A més per a que es pugui utilitzar en criptografia, $e(S, T)$ ha de ser eficientment computable $\forall (S, T) \in G_1 \times G_2$.

Si $G_1 = G_2$ aleshores diem que l'aparellament és simètric. Altrament diem que l'aparellament és asimètric.

Per a que un pairing sigui útil en criptografia cal que sigui computacionalment eficient i que el **problema bilineal de Diffie-Hellman** sigui computacionalment difícil. Aquest problema es pot enunciar com:

Donats un aparellament $e : G_1 \times G_2 \to G_T$, i els punts $P, aP, bP, cP \in G_1$ tals que $e(P, P) \neq 1$, computar $e(P, P)^{abc} \in G_2$.

Els pairings utilitzats actualment en criptografia són els basats en els pairings de Weil i Tate en corbes el·líptiques sobre cossos finits. Aquests pairings són aplicacions bilineals des d'un grup d'una corba el·líptica $E(\mathbb{F}_q)$ cap a un grup multiplicatiu $\mathbb{F}^*_{q^k}$. El paràmetre k és el que s'anomena grau d'immersió de la corba el·líptica.

Definició. (Grau d'immersió): Sigui $E/\mathbb{F}_q$ una corba el·líptica i sigui l un divisor de $N = \#E(\mathbb{F}_q)$ (habitualment l primer gran). S'anomena grau d'immersió de $E/\mathbb{F}_q$ respecte a l, al mínim enter positiu k verificant les condicions equivalents:

- $l \mid (q^k - 1)$
- $\mathbb{F}^*_{q^k}$ conté un subgrup cíclic d'ordre l .

Si l és el major divisor primer de N, k es denomina simplement grau d'immersió de $E/\mathbb{F}_q$. És per això que és molt important escollir de forma apropiada la corba per a que aquesta sigui prou forta als atacs i les operacions siguin computacionalment eficients.

En particular, per a la criptografia basada en la Identitat (la clau pública d'un usuari és el seu propi nom o qualsevol altre atribut lligat a ell mateix) requereix corbes amb un grau d'immersió petit. En particular, les corbes supersingulars (aquelles per les que $p \mid t$) són idònies per a aquest propòsit, ja que aquestes corbes tenen $k \leq 6$. Per contra, les corbes el·líptiques ordinàries amb grau d'immersió petit són una minoria, complicades de trobar i amb una caracterització complexa. En l'apartat 3.2.2, es farà un estudi més detallat d'aquest dos tipus de corbes.

El pairing es considera segur si, agafant logaritmes discrets en els grups $E(\mathbb{F}_q)$ i $\mathbb{F}^*_{q^k}$, són ambdós computacionalment no factibles. Per a un òptim comportament, els paràmetres q i k haurien de ser elegits de tal manera que els dos problemes de logaritmes discrets fossin aproximadament d'igual dificutat quan s'utilitzen els millors algoritmes coneguts, i l'ordre del grup $\#E(\mathbb{F}_q)$ hauria de tenir un factor primer r gran.

Aquests aparellaments de Weil i de Tate són aplicacions bilineals que assignen a un parell de punts (P, Q) de la corba E sobre $\mathbb{F}_q$, una arrel d'ordre l d'una extensió $\mathbb{F}_{q^k}$, on k és el grau d'immersió de la corba i l és un nombre primer i divisor del cardinal de la corba $N = \#E(\mathbb{F}_q)$. Agafant punts de l-torsió com a entrada i com a sortida elements d'un cos finit, els pairings es defineixen utilitzant funcions racionals.

Un càlcul efectiu dels aparellaments, es pot realitar utilitant l'algoritme de Miller [13].

3.2.2 Corbes el·líptiques pairing-friendly

Les corbes el·líptiques amb graus d'immersió petits i subgrups d'ordre primer grans són els ingredients clau per a les implementacions de criptosistemes basats en pairings. En aquest apartat es descriuran àmpliament algunes famílies de corbes el·liptiques que tenen graus d'immersió petits, que han anat sortint durant definicions prèvies, sense aprofundir del tot en el seu significat ni en les seves característiques. Aquestes corbes són les que anomenem *pairing-friendly curves*.

Corbes supersingulars

Una corba el·líptica $E/\mathbb{F}_{p^m}$ es diu que és supersingular si satisfà que

$$p \mid (p^m + 1 - \#E(\mathbb{F}_{p^m})).$$

Totes les corbes el·líptiques supersingulars tenen grau d'immersió $k \leq 6$ i, per tant, són *pairing-friendly.* Per qualsevol corba supersingular, els pairings de Tate o Weil representen un pairing criptogràfic sobre $E[n]$, on n és un

divisor de $\#E(\mathbb{F}_q^*)$. A més a més, un pairing $\hat{e}$ simètric sobre la corba E pot obtenir-se utilitzant l'aplicació

$$\hat{e}(P,Q) = e(P,\Psi(Q)),$$

on e és el pairing de Weil o Tate usual i $\Psi : E(\mathbb{F}_{p^m}) \to E(\mathbb{F}_{p^m})$ és una aplicació de distorsió. El corresponent pairing $\hat{e}$ és conegut com el pairing de Weil (o Tate) modificat.

Corbes ordinàries

Algunes aplicacions com signatures curtes requereixen *pairing-friendly elliptic curves* amb grau d'immersió més gran que 6. A continuació es descriuran mètodes capaços de produir corbes el·líptiques amb un grau d'immersió superior a 6. El primer, el mètode Cocks-Pinch [9] produeix corbes el·líptiques ordinàries (aquelles que no són supersingulars) amb graus d'immersió arbitraris. La segona construcció, la de Barreto-Naehrig [2], produeix corbes el·líptiques amb grau d'immersió $k = 12$ i d'ordre primer.

Mètode Cocks-Pinch El mètode de Cocks-Pinch produeix corbes el·líptiques ordinàries que tenen un grau d'immersió arbitrari. És considerat el mètode més flexible per construir corbes ordinàries *pairing-friendly*.

Suposem que volem construir una corba E sobre $\mathbb{F}_q$ de grau d'immersió k i que el seu cardinal factoritza en un primer gran l. Aquest mètode es basa en què si n és el cardinal de la corba, aleshores $l \mid n = (p + 1 - t)$ i $l \mid (p^k - 1)$.

Corbes Barreto-Naehrig La família de corbes el·líptiques de Barreto-Naehrig [2] tenen un grau d'immersió $k = 12$ i a més a més tenen ordre primer. Per a generar-les, es consideren els polinomis:

$$\begin{aligned} N(x) &= 36x^4 + 36x^3 + 18x^2 + 6x + 1 \\ P(x) &= 36x^4 + 36x^3 + 24x^2 + 6x + 1 \end{aligned}$$

i s'agafa un valor de x_0 per al que tant $n = N(x_0)$ com $p = P(x_0)$ siguin primers. (Un exemple podria ser $x_0 = 82$ per al que tindríem $n = 1647609109$ i $p = 1647649453$, ambdós primers).

Les corbes que es construeixen amb aquest mètode, tenen equació $y^2 = x^3+b$. Així, busquem ara un valor $b \in \mathbb{F}_p$ per al que $b+1$ sigui un residu quadràtic en $\mathbb{F}_p$, i el punt $Q = (1, \sqrt{b+1})$ en la corba el·líptica $E : y^2 = x^3 + b$ satisfaci que $n \cdot Q = \mathcal{O}$. El procediment de cerca podria ser tant simple com començar per $b = 1$ i incrementar b gradualment fins que el valor adequat es trobi. Per a tal valor b, la corba $E/\mathbb{F}_p$ que ve donada per l'equació $y^2 = x^3 + b$ té n punts, grau d'immersió $k = 12$ i el punt $Q = (1, \sqrt{b+1})$ pot ser agafat com a punt base.

3.3 Criptografia basada en la identitat

Per evitar problemes amb l'autenticació de les claus públiques (certificats i autoritats de certificació) que planteja la criptografia de clau pública clàssica de Shamir, ell mateix l'any 1984 va proposar un nou paradigma: la criptografia basada en la Identitat [18]; en la qual la clau pública d'un usuari és el seu propi nom o qualsevol altre atribut lligat a ell mateix. Exemples d'aquests atributs poden ser les direccions de correu electrònic, números de telèfon, IP's o noms de dominis entre altres. Aquests atributs, a part de ser únics (no es poden repetir o duplicar), cada un pertany a una entitat en concret, que serà la que estarà en una banda del canal de comunicació.

Aquest esquema el que permet és comunicar dos usuaris de manera segura i verificar cada un la signatura de l'altre sense haver de fer comprovacions de certificats, sense haver de mantenir directoris de claus i sense haver d'utilitzar el servei d'una tercera part. El que sí que assumeix l'esquema, és l'existència de centres de generació de claus, de confiança, la finalitat dels quals és donar a cada usuari una "targeta intel·ligent personalitzada" quan s'uneix per primera vegada al sistema. La informació que conté aquesta targeta permet a l'usuari signar i encriptar els missatges que envia i desencriptar i verificar els missatges que rep d'una manera independent, sense haver de tenir en compte la identitat de l'altra part.

L'esquema és ideal per grups tancats d'usuaris com poden ser executius d'una multinacional o les sucursals d'un gran banc, ja que l'oficina central de la companyia pot servir com a generador central de claus en el que tothom confia. Aquest esquema pot ser la base per a un nou tipus de targeta d'identificació personal amb la qual, cadascú, electrònicament pot firmar xecs,

rebuts bancaris, documents legals i correus electrònics.

3.3.1 Esquema bàsic de funcionament

A partir d'ara anomenarem al generador de claus privades **PKG** (Private Key Generator). Abans de que les operacions puguin començar, el PKG considera un generador pk_{PKG} del grup on treballarem, un enter s que serà la clau secreta global i una clau pública mestra sk_{PKG} que es troba a partir de s i pk_{PKG}. Per operar, el PKG el primer que fa és publicar sk_{PKG} als usuaris dels seus serveis, per a que així tots puguin calcular la clau pública corresponent a una entitat combinant la clau pública mestra amb la cadena identificativa de l'entitat. Per obtenir la corresponent clau privada, la part autoritzada a utilitzar-la, contacta amb el PKG per a que utilitzi s per a generar la clau privada per a l'entitat ID.

Encriptació basada en la Identitat (IBE)

A continuació s'exposa un exemple teòric en el que es pot veure la finalitat de l'encriptació basada en la Identitat i com treballa a nivell teòric:

1. L'Alice vol enviar un missatge M al Bob. Ella utilitza la identitat del Bob que anomenarem ID_{BOB} i la clau pública mestra sk_{PKG} per encriptar M, obtenint un missatge encriptat C. L'Alice li envia al Bob el missatge C. Es veu com, abans de l'encriptació, l'Alice ja coneixia ID_{BOB} i $Q_{ID_{BOB}}$, així doncs, no ha estat necessària l'aportació d'informació/preparació d'en Bob.

2. En Bob rep el missatge de l'Alice. En algunes implementacions s'assumeix que C s'envia amb instruccions per contactar amb el PKG per agafar la clau privada requerida per desencriptar. En Bob s'autentifica amb el PKG, bàsicament enviant la suficient informació/prova de que a ID_{BOB} li pertany a ell. Un cop fet, el PKG li transmet al Bob la seva clau privada $D_{ID_{BOB}}$ per un canal segur.

3. En Bob desencripta C utilitzant la seva clau privada $D_{ID_{BOB}}$ per recuperar el missatge original M.

Una variació podria ser que el PKG desencriptés C per a en Bob i ja li transmitís a ell després d'autentificar-se, per a que el sistema encara fós més

transparent.

En la figura 3.3 es pot observar de manera senzilla, sense entrar en detalls, com funciona el procés en cas de voler enviar un correu electrònic. En aquest cas l'Alice és la que vol enviar el missatge al Bob i utilitza la seva identitat (el seu correu electrònic).

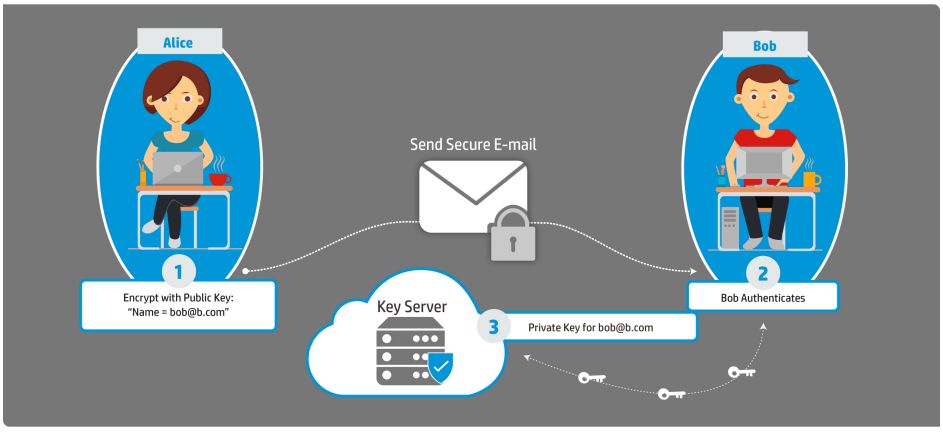

Figura 3.3: Procés de l'enviament d'un missatge utilitzant IBE

3.3.2 Pros i contres de l'Encriptació Basada en la Identitat (IBE)

- **PROS**
  - No es necessiten certificats. A partir de la identitat d'una persona (com pot ser el seu mail, el seu nom o alguna altra dada que l'identifiqui de forma única), es pot obtenir la seva clau pública.
  - No es necessita una presa de contacte prèvia abans de la recepció/enviament de missatges per part d'ambdues parts.
  - Les claus caduquen, pel que no necessiten ser anul·lades. En un sistema de clau pública tradicional, les claus hauran de ser anul·lades si són compromeses.
  - Hi ha baixa vulnerabilitat per spam.

  - Permet l'expiració automàtica, fent que els missatges no siguin "llegibles" passat un cert temps.

- **CONTRES**

  - Requereix un servidor centralitzat. L'enfoc centralitzat implica que algunes claus hauran de ser creades i mantingudes "sota custodia" i tenen, per tant, un risc més alt de divulgació.
  - Requereix un canal segur entre emisor/receptor del missatge i el servidor IBE per poder transmetre de manera segura la clau privada, requerida pel receptor en el moment de desencriptar. En cas de voler delegar feina (de l'usuari cap al servidor central) podem també fer que sigui el servidor central el que desencripti el missatge abans de guardar-lo en la base de dades i que es demani una validació per part de l'usuari receptor per a saber que el missatge original anava destinat cap a ell.

Ús d'aparellaments en IBE

A continuació s'enumeraran les parts principals de l'esquema d'encriptació basada en la identitat [1, 17] proposat per Boneh-Franklin, però pensant ara ja, amb una implementació en la que es requeriran aparellaments bilineals per a que tot funcioni correctament:

- Paràmetres del sistema:
  - Pairing $e : G_1 \times G_1 \to G_T$
  - G_1 un grup additiu.
  - G_T és un grup multiplicatiu
- El **PKG** (Private Key Generator) disposarà de :
  - Un generador P_k de G_1.
  - Una clau secreta global s (és un enter), una clau pública mestra $S_k = s \cdot P_k$
- Es disposarà en tot moment d'una funció hash $h : G_T \to \{0,1\}^n$.

- El PKG generarà **per a cada usuari** les següents claus:
  - Un usuari A té una clau pública $Q_A \in G_1$
  - La clau privada de l'usuari A és $D_A = s \cdot Q_A$
- Si volem, per exemple, **encriptar** un missatge m per enviar-li a l'usuari A seguirem els següents passos:
  - Escollir un enter aleatori que anomenarem r.
  - Calcular la tupla $(U, V) = (r \cdot P_k, m \oplus h(e(Q_A, S_k)^r))$.
  - Enviar la tupla (U, V) a l'usuari A.
- En el moment que A vol desencriptar el missatge que se li ha enviat, usant la seva clau privada D_A, farà:
  - Recupera el missatge m calculant: $m = V \oplus h(e(D_A, U))$.
    - Cal notar que és compleix la següent propietat:

$$e(D_A, U) = e(s \cdot Q_A, r \cdot P_k) = e(Q_A, s \cdot P_k)^r = e(Q_A, S_k)^r$$

3.3.3 Signatura basada en la Identitat (IBS)

En aquesta secció es descriurà el procediment que es duu a terme en la signatura basada en la identitat que va proposar Shamir per permetre que els usuaris verifiquessin una signatura utilitzant informació pública com pot ser l'identificador d'un usuari.

1. L'Alice s'autentifica amb el PKG i rep la seva clau privada $D_{ID_{ALICE}}$.
2. Utilitzant la seva clau privada $D_{ID_{ALICE}}$, l'Alice genera una signatura σ per a un missatge M i li transmet a en Bob.
3. Després de rebre M i σ de l'Alice, en Bob comprova que σ sigui autèntica en M utilitzant la identitat de l'Alice ID_{ALICE} i la clau pública mestra sk_{PKG}. Si és autèntica envia "Accept". En cas contrari envia "Reject". Cal veure que en Bob no necessita tenir cap tipus de certificat de l'Alice.

Aquesta va ser la forma que va proposar Shamir d'utilitzar l'algoritme RSA per a la firma electrònica, però es va haver d'esperar fins al 2001 per a que Boneh i Franklin proposessin sistemes per aconseguir sistemes de xifrat basat en la identitat o IBE (Identity-Based Encryption).

Signatura de Boneh, Lynn i Shacham

En aquesta secció es descriuran les principals característiques de la signatura descrita per Boneh, Lynn i Shacham [4], que permet que un usuari verifiqui si una signatura és autèntica. Aquest esquema té signatures amb 170 bits de longitud envers els 320 bits que tenen les signatures DSA.

Paràmetres:

- Tenim un pairing $e : G_1 \times G_1 \to G_2$
- G_1 i G_2 són grups d'ordre primer r
- P és un generador de G_1
- h és la funció de codificació i $h : \{0,1\}^n \to G_1$

Algoritme de generació de signatura

1. INPUT: Paràmetres anteriors (G_1, G_2, e, P), la clau privada d i el missatge M
2. OUTPUT: El missatge M amb la signatura σ
3. Procediment:
   (a) Calcular el hash del missatge: $H = h(M) \in G_1$.
   (b) Calcular $\sigma = dH \in G_1$
   (c) Retornar M i σ

Algoritme de verificació de la signatura

1. INPUT: Els paràmetres (G_1, G_2, e, P), la clau pública Q i M amb σ
2. OUTPUT: Acceptació o rebuig de la signatura
3. Procediment:
   (a) Calcular el hash del missatge : $H = h(M) \in G_1$
   (b) Si $e(\sigma, P) = e(H, Q)$ aleshores retorna "signatura acceptada"

Cal tenir present que $e(\sigma, P) = e(dH, P) = e(H, dP) = e(H, Q)$

Capítol 4

IBS vs IBOOS

En aquest capítol es farà una breu però detallada descripció del paper que es farà servir de referència per a la implementació del projecte, la qual consisteix en un sistema compost per la utilització de dos tipus diferents de signatures. L'article titulat «A Novel ID-based Authentication Framework with Adaptative Privacy Preservation for VANETs» i escrit per Huang Lu i Jie Li de la «University of Tsukuba» i Mohsen Guizani de la «Qatar University» [10], parla de la combinació del que anomenem IBS (Identity-based signature) i, per altra banda, IBOOS (Identity-based Online/Offline signature). La informació d'aquest article, serà complementada amb la informació d'un altre estudi realitzart per R.Nasreen Salma, N.Alangudi Balaji i el Dr.R.Sukumar del «Sethu Institute of Thecnology Pulloor, Kariapatti, India» [15], que també tenen un framework similar per tal de poder trobar una descripció més acurada.

4.1 Introducció

Les principals propostes que es fan en l'article [10] són les següents:

- Proposen un framework eficient i segur per realitzar autenticacions preservant la privacitat en les VANETs.

- El framework d'autenticació que es proposa no restringeix en cap moment la solució a la utilització únicament dels esquemes IBS i IBOOS existents, se'n poden utilitzar de nous.

- A diferència d'altres esquemes, es carrega a cada vehicle el conjunt de RSUs regionals, perque el nombre de RSUs és relativament petit i el nombre de RSUs no varia molt freqüentment.

- S'utilitzarà l'esquema IBS per a les autenticacions V2R (Vehicle-to-Roadside) i R2V (Roadside-to-Vehicle), i l'esquema IBOOS per a l'autenticació V2V (Vehicle-to-Vehicle).

4.2 Descripció del sistema i objectius

Estructura de la xarxa i components

Com hem explicat amb anterioritat en 2.1.1, els components bàsics de la xarxa són els RSUs, els vehicles i RTAs en els que es pot confiar. Les possibles combinacions, recordem que eren V2V, R2V i V2R. Altres comunicacions que tenim són les inter-RSU i les RSU-to-RTA.

Tots els vehicles utilitzen canals de ràdio simètrics i els vehicles tenen montades unes unitats segures per emmagatzemar dades sensibles (OBUs). Els candidats de RTA són els fabricants d'automòbils, un tercer autenticat de l'estat, etc. Les principals responsabilitats del RTA, que no s'ha aprofundit en 2.1.1, són les següents:

- Genera tot el material criptogràfic (claus) per als RSU i per als vehicles de la regió, i els distribueix per canals segurs.

- Maneja la CRL (llista de vehicles dels quals se n'ha revocat la participació), l'actualitza i informa a la xarxa per aïllar als vehicles del possible perill.

- Si un missatge és enviat per un vehicle que té un problema en carretera, el RTA ha de localitzar i identificar l'emisor per poder resoldre la disputa.

- Els vehicles procedents d'altres regions han de poder autenticar-se per tots els RTA per on van passant, pel que han de tenir informació creuada entre ells.

4.3 Descripció del sistema proposat

El mecanisme bàsic de funcionament seria el següent:

1. Els vehicles es registren inicialment al RTA (Regional Trusted Authority) amb la seva informació d'usuari.

2. Aleshores passen dues coses:

   (a) Els vehicles poden començar a comunicar-se amb els RSU (R2V i V2R) i amb altres vehicles (V2V).

   (b) El RSU, per altra banda, comprova la identitat del vehicle amb l'ajuda del RTA pe veure si és vàlid.

3. Un cop està tot comprovat, els vehicles es comuniquen amb els RSU per poder comunicar-se amb altres entitats de manera segura.

Però aquest mecanisme bàsic de funcionament es veu alterat en el moment que ens adonem que podem reduir el temps de verificació d'autenticitat i que la fase de registre es pot produir sense compromisos a l'inici del registre. Com ja hem comentat, s'utilitzarà l'esquema IBS per a les autenticacions V2R (Vehicle-to-Roadside) i R2V (Roadside-to-Vehicle), i l'esquema IBOOS per a l'autenticació V2V (Vehicle-to-Vehicle).

4.4 Funcionament bàsic del sistema proposat

En aquesta secció, primer de tot desglossarem en fases cada una de les dues signatures utilitzades. A continuació, es presentaran els diagrames de fluxe per veure el funcionament del sistema d'una forma més visual junt amb els components implicats en cada part. Però primer, es presentaran els passos i les descripcions d'aquests per a les dues signatures:

Signatura basada en la Identitat (IBS) L'esquema de la generació de la signatura amb IBS consisteix en quatre passos:

- **Setup**: Registre dels vehicles amb el RTA utilitzant les seves identitats. L'**RTA** computa la clau mestra s i els valors públics *params* i li dóna tots els paràmetres als vehicles.

- **Extraction**: Donat un ID en format string, l'algoritme genera una clau privada associada al ID amb la clau mestra.
- **Signature**: Donat un missatge, l'algoritme genera una signatura.
- **Verification**: Donat un ID, un missatge i una signatura, l'algoritme de verificació treu «accept» si la signatura és vàlida. Si no es vàlida treu «reject».

Signatura basada en la Identitat Online/Offline (IBOOS) L'esquema de la generació de la signatura amb IBOOS consisteix en 5 passos (amb dos signatures):

- **Setup**: El mateix que el setup de l'esquema IBS.
- **Extraction**: El mateix que el extraction de l'esquema IBS.
- **Offline signing**: Donats els paràmetres públics, l'algoritme genera una **signatura offline.**
- **Online signing**: Donada la clau privada, la signatura offline i el missatge, l'algoritme genera la signatura online del missatge.
- **Verification**: Donat un ID, un missatge i la signatura online, l'algoritme de verificació treu «accept» si la signatura és vàlida. Si no es vàlida treu «reject».

En el proper apartat s'explicarà a nivell més teòric i matemàtic com funciona cada un dels components.

En la figura 4.1 es mostra un esquema bàsic on es poden veure més clarament les verificacions, des del registre del vehicle fins a la comunicació entre vehicles on s'utilitza IBOOS. Ara anem a parlar amb més detall de com funciona l'algoritme. En la fase de registre, els vehicles es registren a la RTA utilitzant les seves identitats reals. Inicialment, el RTA s'assumeix que són autoritats en les que es pot confiar i que són «cross-certified» per a extendre l'ús de paràmetres que ja estan certificats i, per tant, reduir la càrrega en el registre i en la verificació. Aleshores, els venicles agafen els paràmetres certificats del RTA, els quals han de ser extrets per a la generació de la signatura. La signatura és verificada llavors per autenticar els usuaris.

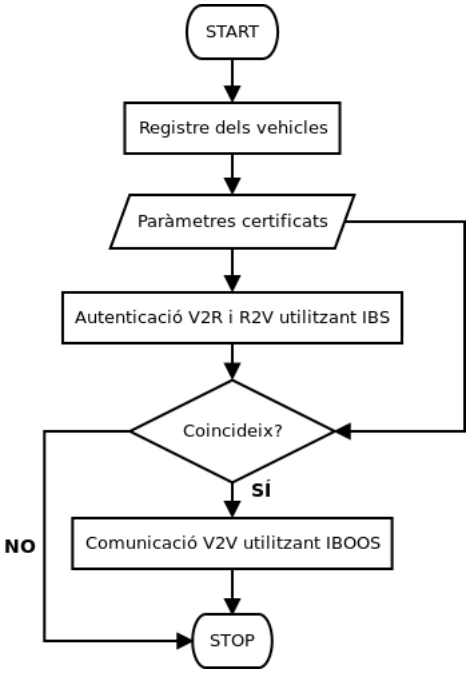

Figura 4.1: Procés de verificació de les signatures

En el procés de verificació de la signatura, els paràmetres són extrets per el RSU utilitzant l'esquema IBS, tal i com s'han explicat anteriorment els passos. Les comunicacions entre vehicles, com es pot veure en la figura 4.2, tenen lloc utilitzant l'esquema IBOOS.

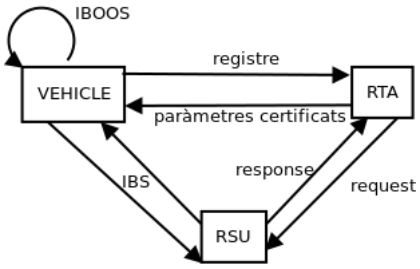

Figura 4.2: Registre de vehicles i procés d'autenticació

4.5 Esquema basat en la identitat Online/Offline

A continuació s'explicarà més detalladament l'esquema basat en la identitat Online/Offline que s'utilitzarà en la implementació, el qual s'ha extret de [15]. Com hem explicat en 4.4, aquest esquema també està format per 5 components fins arribar a la verificació final. Per tal de deixar tot perfectament detallat, explicarem component a component amb tots els càlculs necessaris ampliant :

1. Setup. En el nostre cas, el RTA el que farà serà:
   (a) Seleccionar un generador $g \in \mathbb{G}$, on $(\mathbb{G}, *)$ és un grup multiplicatiu d'ordre primer q.
   (b) Seleccionar un $x \in \mathbb{Z}_q^*$ random que serà la nostra «master secret key».
   (c) Calcular $X = g^x \in \mathbb{G}$.
   (d) Considerar una funció hash $H_1 : \mathbb{G} \times \{0,1\}^* \to \mathbb{Z}_q^*$.
   (e) Considerar una funció hash $H_2 : \mathbb{G} \times \mathbb{G} \times \{0,1\}^* \to \mathbb{Z}_q^*$.
2. Extract. Per generar una clau secreta per a la identitat de l'usuari **ID**, el RTA el que fa és:
   (a) Seleccionar de forma random $r \in \mathbb{Z}_q^*$.
   (b) Computar $R = g^r \in \mathbb{G}$.
   (c) Computar $s = r + H_1(R, ID)x \mod q$

   La clau secreta de l'usuari serà (R, s). Per a comprovar la correctesa de la clau secreta podem dur a terme el següent càlcul:
   $$g^s = R * X^{H_1(R,ID)}$$
3. Signatura Offline. La signatura offline no requereix el coneixement previ ni del missatge ni de la clau secreta, pel que pot ser generada pel RTA enlloc d'haver-se de generar per a cada vehicle. El que signa, en aquest cas el nostre RTA, computa:
   $$\hat{Y}_i = g^{2^i} \in \mathbb{G} \; for \; i = 0, ..., |q| - 1$$

4. Signatura Online. La signatura online sí que la durà a terme el vehicle. En aquest cas el que farà serà:

   (a) Seleccionar un $y \in \mathbb{Z}_q^*$ random. Tindrem que $y[i]$ és el i-th bit de y

   (b) Definirem $\mathcal{Y} \subset \{1, ..., |q|\}$ com al conjunt d'indexos que compleixen que $y[i] = 1$

   (c) Calcularem:

      i. $Y = \prod_{i \in \mathcal{Y}} \hat{Y}_{i-1} \in \mathbb{G}$
      ii. $h = H_2(Y, R, m)$ on m és el missatge
      iii. $z = y + h\, s \, mod \, q$

   La signatura és (Y, R, z)

5. Verificació. Per a verificar la signatura (Y, R, z) per al misatge m i per a la identitat ID, el verificador el primer que farà serà calcular $h = H_2(Y, R, m)$ i comprovar que es compleixi la següent igualtat:

$$g^z = Y * R^h * X^{hH_1(R,ID)}$$

   Aleshores, s'acceptarà si es compleix la igualtat i, en cas contrari, es rebutjarà.

Capítol 5

Implementació

En aquest capítol parlarem del treball d'implementació que s'ha dut a terme per a mostrar d'una forma més dinàmica com funciona la comunicació entre els diferents elements que formen la xarxa vehicular. Primer de tot farem una descripció més teòrica i després procedirem a parlar una mica en detall del codi que s'ha fet. Cal recordar que l'esquema que s'intenta mostrar mitjançant aquesta implementació és l'esquema explicat en el capítol anterior en el que s'utilitza IBs per a l'autenticació entre RSU i vehicle i IBOOS per a l'autenticació entre vehicles.

5.1 Introducció i descripció prèvia

En aquesta secció parlarem de l'estructura de la xarxa seguida durant la implementació i quines estratègies s'han pres. Previ a això cal tenir en compte unes consideracions que s'han tingut a l'hora de definir una substructura:

- Disposarem de tres vehicles únicament dins de la xarxa.
- Disposarem d'un RTA, que farà de RTA per a la regió de Catalunya.
- Disposarem d'un RSU per a fer proves que he definit, per posar un example, com al RSU que es troba en el tram de carretera de la C-14 de Ponts a Artesa de Segre com a exemple. En definitiva, no es crearà una estructura completa sinó que es mostrarà que passa en un espai petit de territori, on el RSU d'aquest tram de carretera es registrarà al RTA de Catalunya i els tres vehicles, a l'entrar en aquest tram de carretera,

es registraran amb el RTA primer i seguidament s'autenticaran amb el RSU corresponent.

IBS

Com ja hem especificat en el capítol anterior, en l'esquema, utilitzem la signatura basada en la identitat per a l'autenticació prèvia a la comunicació entre el RSU i el vehicle. Específicament, l'esquema utilitzat per a la signatura basada en la identitat en aquest cas és la signatura de Boneh, Lynn i Shacham. L'esquema en detall el podem trobar en 3.3.3.

IBOOS

Per altra banda, l'esquema que s'ha utilitzat per representar l'esquema Online/Offline basat en la identitat s'ha extret de [PAPER] i explicat en més detall en 4.5. Aquest esquema és el que s'utilitzarà per verificar la validesa d'un vehicle abans d'establir una connexió entre dos vehicles (V2V).

5.2 Implementació

La idea bàsica per a la implementació ha estat seguir l'esquema bàsic de funcionament que s'ha presentat en les figures 4.1 i 4.2 de la secció 4.4. A continuació es presentaran algunes de les coses més importants a tenir en compte per tal d'entendre com funciona la llibreria jPBc [11] utilitzada i quins components són els més importants i els que hem decidit utilitzar per a implementar de manera senzilla l'esquema.

5.2.1 Tipus de corbes suportades per la jPBC

La llibreria jPBC suporta els següents tipus de corbes:

1. Tipus A: Són corbes supersingulars d'equació $y^2 = x^3 + ax + b,\ a \in \mathbb{F}_q$ amb grau d'inmersió $k = 2$.

2. Tipus A1: Són les mateixes corbes utilitzades que pel tipus A. S'utilitzen quan el el criptosistema necessita que el cardinal de la corba sigui un nombre específic, per exemple $N = p{\cdot}q$ amb p i q nombres primers grans, fent que N sigui difícil de factoritzar.

3. Tipus D: Són cobes amb grau d'inmersió $k = 6$. Es poden construir amb el mètode CM.

4. Tipus E: Són corbes amb grau d'inmersió $k = 1$ construides utilitzant el mètode CM. Tots els còmputs necessaris per fer el pairing es poden realitzar sobre $\mathbb{F}_q$.

5. Tipus F: Són una família de corbes amb grau d'inmersió $k = 12$. Van ser proposades per Barreto i Naehrig utilitzant també el mètode CM i considerant els polinomis ciclotòmics. [2]

6. Tipus G: Són corbes amb un grau d'inmersió $k = 10$. Van ser proposades per Freeman. [8]

5.2.2 Operacions de la jPBC bàsiques per a pairings

En aquesta secció descriurem els mètodes bàsics de la jPBC per a treballar amb pairings:

1. **Utilitzar les funcions implementades en C quan sigui possible**. La llibreria jPBC es basa en la llibreria PBC que va desenvolupar Ben Lynn[11] utilitzant el llenguatge de programació C. La llibreria jPBC importa tot el codi de la llibreria PBC i a vegades es dóna el cas de que la jPBC crida a la llibreria PBC degut a que C és més òptim que Java en algunes operacions. Degut a que Java és un llenguatge «lent» respecte d'altres, com per exemple C, s'ha donat la possibilitat a la llibreria jPBC de tenir operacions de pre-processament de les operacions d'exponenciació i de pairings, molt útils quan aquestes operacions s'han de repetir al llarg del codi. Així doncs per utilitzar les funcions que provenen de la PBC s'ha d'activar de la següent forma:

```
PairingFactory.getInstance().setUsePBCWhenPossible(true);
```

2. **Generació aleatòria** d'una corba de tipus A, que és el tipus que s'ha utilitzat en la implementació:

```
PairingParametersGenerator ppg = new
TypeACurveGenerator(ThreeParty.rBits,ThreeParty.qBits);
Pairing e = PairingFactory.getPairing(ppg.generate(),
    null);
```

3. **Càrrega del fitxer amb els paràmetres de l'aparellament.** Partim de l'existència d'un fitxer de configuració com podria ser el següent, amb la configuració $rBits = 128$ i $qBits = 256$, que és l'utilitzat en la implementació:

```
type a
q 78910235960302648236085819294375443237752251445803596769379
72552555413508927
r 170141183460469231731687303715917660161
h 463792682966947442862892955073819492 48
exp1 25
exp2 127
sign0 1
sign1 1
```

   que s'ha generat volcant en un fitxer el resultat de ppg.generate() del codi del punt anterior.
   L'avantatge que ens dóna això és el poder disposar en tot moment d'un mateix pairing, sense que vagi canviant en cada execució del programa. Per carregar-lo al programa el que fem és el següent:

```
Pairing e = PairingFactory.getPairing("params.params")
```

   on *param.params* és el fitxer amb la forma descrita anteriorment.

4. **Obtenir els cossos** (estructures algebraiques) que intervenen en un pairing:

```
Field Zr = e.getZr();
Field G1 = e.getG1();
Field G2 = e.getG2();
Field H  = e.getGT();
```

5. **Obtenir un element aleatori d'un camp.** En el codi següent es mostra com obtenir un element aleatori de $\mathbb{G}_1$:

```
Element el = G1.newRandomElement();
```

6. **Fer que un element sigui inmutable**, és a dir, que el seu valor no varïi:

```
el = el.getImmutable();
```

7. **Multiplicació d'un element per un enter:**

```
int integer = 4;
Element p = e.getG2().newRandomElement().getImmutable();
p = p.mul(new BigInteger(String.valueOf(integer)))
```

8. **Obtenir un pairing de dos elements.** La interfície Pairing ens dóna mètodes per aplicar la funció pairing. Donats dos elements Element $in1$ i $in2$, cada un pertanyent a l'estructura algebraica adequada, el seu pairing pot ser computat invocant el mètode pairing amb $in1$ i $in2$ com a imputs de la següent forma:

```
Element out = pairing.pairing(in1, in2);
```

9. **Potència d'un element a** z, on z és un element de $\mathbb{Z}_r$:

```
Element e = G1.newRandomElement();
Element z = pairing.getZr().newRandomElement();
e.powZn(z);
```

10. **Obtenir la representació en bytes d'un element:**

```
byte[] bytes = e.toBytes();
```

11. **Recuperació d'un element a partir dels bytes:**

```
Element U = e.getG1().newElementFromBytes(bytes)
```

12. **Creació d'un BigInteger a partir d'un element**:

```
BigInteger bi = new BigInteger(element.toBytes())
```

5.2.3 Implementació del sistema IBS/IBOOS proposat

El sistema IBS/IBOOS com el nom indica i com hem explicat amb anterioritat, està basat en dos esquemes diferents: IBS per a les comunicacions V2R i IBOOS per a les comunicaciones V2V entre vehicles. Per simplicar la tasca d'entendre la implementació i com funciona, dividirem l'explicació en comentar cada un dels esquemes per separat.

Base de dades utilitzada

Abans de començar a explicar en més profunditat el codi que s'ha empleat per a dur a terme la implementació, cal deixar ben definida la base de dades utilitzada per a emmagatzemar tota la informació necessària.

S'ha utilitzat una connexió en local amb MySQL on tenim la nostra base de dades que hem anomenat **VanetDB** la qual està formada per 4 taules per

emmagatzemar les dades. Aquestes taules i la seva composició d'atributs són les següents:

1. Taula **RTA**. Aquesta taula és l'encarregada d'emmagatzemar les claus dels RTA. Està formada pels següents atributs :

   (a) RTAID. Id que s'autogenera per a l'RTA.

   (b) PublicKey. Clau pública del RTA.

   (c) PrivateKey. Clau privada del RTA.

   (d) SecretKey. Clau secreta del RTA.

2. Taula **RSU**. Aquesta taula és l'encarregada d'emmagatzemar les claus dels RSU. Està formada pels següents atributs:

   (a) idRSU. Id que s'autogenera per a l'RSU.

   (b) RSUName. Nom que se li assigna l'RSU segons la zona a la que pertanyi.

   (c) RSUPublicKey. Clau pública del RSU.

   (d) RSUPrivateKey. Clau privada del RSU.

3. Taula **VEHICLE**. Aquesta taula és l'encarregada d'emmagatzemar les dades dels vehicles. Està formada pels següents atributs:

   (a) licensePlate. Identificador únic del vehicle, en aquest cas la matrícula.

   (b) brand. Marca del vehicle.

   (c) model. Model del vehicle.

4. Taule **KEY**. Aqueta taula és l'encarregada d'emmagatzemar totes les claus dels vehicles. Està formada pels següents atributs:

   (a) licensePlate. Identificador únic del vehicle, en aquest cas la matrícula.

   (b) PublicKey. Clau pública del vehicle per a l'encriptació de missatges.

(c) PrivateKey. Clau privada del vehicle per a l'encriptació de missatges.

(d) SecretKey. Clau pública del vehicle per a l'autenticació.

(e) PublicAuthenticationKey. Clau pública del vehicle per a l'autenticació.

Implementació de la comunicació V2R utilitzant IBS

- Quan un vehicle es **registra**, totes les seves dades es guarden en una taula de la nostra base de dades. El procés de registre consisteix en generar totes les claus necessàries per les futures comunicacions. El RTA genera una clau privada i una clau pública per a cada vehicle que es registra. Un cop registrat el vehicle i generades les seves claus, aquestes són enviades per un canal segur del RTA al vehicle.

```
1  private Pair generateKeys() {
2
3      Element publicKey = e.getG1().newRandomElement().
           getImmutable();
4      System.out.println("Public key generated ---> " +
           publicKey);
5
6      Element privateKey = publicKey.mul(new BigInteger(
7              String.valueOf(this.s))).getImmutable();
8
9      System.out.println("Private Key generated ---> " +
           privateKey + "\n");
10
11     return new Pair(publicKey, privateKey);
12 }
```

- A la zona on estem, seguidament ens «registrem» amb l'RSU més proper per la zona on estem.

```
1 RSU rsu = new RSU("RSU-C14-1", rta);
2 rta.registerRSU(rsu);
3 v1.linkToRSU(rsu);
```

- Aleshores utilitzant la signatura de Boneh, Lynn i Shacham (explicada en detall en 3.3.3), generem la signatura i li enviem al RSU que serà el que ens haurà de donar per vàlids o no, verificant la signatura.

```
1  public Pair generateIBSSignature() throws
       UnsupportedEncodingException {
2      String M = "Hello message";
3      Pairing e = PairingFactory.getPairing("params.params"
           );
4      Element hash = hash(M);
5      Pair keys = (Pair) this.params.getSecond();
6      int privateKey = (int) (keys.getSecond());
7
8      Element sigma = e.getG1().newRandomElement();
9
10     sigma.set(hash.mul(new BigInteger(String.valueOf(
           privateKey))).getImmutable());
11     Element p = e.pairing(sigma, e.getG1().
           newRandomElement());
12     return new Pair(sigma, M);
13 }
```

Com es pot observar, es necessita una funció hash que anomenem hashFunction que el que fa és el següent:

```
1  private BigInteger hashFunction(Element Y, Element R,
       String m) {
2      Pairing e = PairingFactory.getPairing("params.params"
           );
3      Element el = e.getZr().newRandomElement();
4      byte[] bytes = m.getBytes();
5      el.setFromHash(bytes, 0, bytes.length).getImmutable()
           ;
6      Element hashElement = R.mulZn(el).mulZn(Y).
           getImmutable();
7      return new BigInteger(hashElement.toBytes());
8  }
```

- Un cop el RSU rep la signatura procedent del vehicle la verifica per veure si el vehicle és qui diu ser.

```
1  public boolean checkIfSignatureIsCorrect(Pair ep){
2
3      Pairing e = PairingFactory.getPairing("params.params"
           );
4      Element hash = hash((String) ep.getSecond());
5      Element vehiclePublicKey = getPublicAuthKey();
6      Element RTAPublicKey = this.rsu.getRTA().
           getRTAPublicKey();
7
```

```
 8      Element sigma = ((Element) ep.getFirst());
 9      Element p1 = e.pairing(sigma, RTAPublicKey);
10      Element p2 = e.pairing(hash, vehiclePublicKey);
11      System.out.println("First part: " + p1);
12      System.out.println("Second part: " + p2);
13
14      return p1.isEqual(p2);
15  }
```

Implementació de la comunicació V2V utilitzant IBOOS

Com ja hem descrit en la secció anterior 4.5, l'esquema IBOOS es basa en 5 components que hem dividit en 5 funcions dins del nostre codi on el que fem és:

1. El RTA serà l'encarregat de dur a terme els tres primers components, que són el Setup, el Extract i la signatura Offline. Aquesta última pot realitzar-se pel RTA ja que no necessitem en cap moment conèixer ni el contingut del missatge ni la identitat de cap vehicle. Per a no sobrecarregar de treball al vehicle he decidit cedir-li aquesta feina al RTA.

   (a) **Setup**. En aquest mètòde fixem els valors als elements g,x,q i X

   ```
   1  public void iboosSetup() {
   2      this.g = e.getZr().newRandomElement();
   3      this.gg = g.duplicate().getImmutable();
   4      this.x = e.getZr().newRandomElement().
              toBigInteger();
   5      this.q = new BigInteger(this.r);
   6      this.X = g.pow(x).getImmutable();
   7      System.out.println("X = g^x = " + X);
   8  }
   ```

   (b) **Extract**. Generem la R i la s que formaran la clau secreta de l'usuari.

   ```
   1  public Pair generateSecretKey(String id) {
   2
   3      BigInteger ra = e.getZr().newRandomElement().
              toBigInteger();
   4      Element R = g.pow(ra);
   5      Element R2 = R.duplicate().getImmutable();
   6      Element R3 = R.duplicate().getImmutable();
   ```

```
7
8       BigInteger ss = (ra.add(IBOOSHash(R2, id)).
            multiply(x)).mod(q.subtract(BigInteger.ONE));
9       Element e1 = g.pow(ss);
10      Element e2 = R.mul(X.pow(IBOOSHash(R2, id)));
11
12      /*The following code lines are the checking
            process of the correctness
13      of the calculated element values*/
14      System.out.println("\nFirst part of equality: " +
            e1);
15      System.out.println("Second part of equality: " +
            e2);
16      return new Pair(R3, ss);
17      }
```

Com es pot veure, es necessita una funció de Hash que hem anomenat IBOOSHash i que fa el següent:

```
1  public BigInteger IBOOSHash(Element R, String id) {
2      Element el = e.getZr().newRandomElement();
3      byte[] bytes = id.getBytes();
4      el.setFromHash(bytes, 0, bytes.length).
           getImmutable();
5      Element hashElement = R.mulZn(el).getImmutable();
6      return new BigInteger(hashElement.toBytes()).mod(
           this.q);
7  }
```

(c) **Signatura Offline**. Generem $\hat{Y}$ a partir de g:

```
1  public List offlineSign() {
2      List<Element> Y = new ArrayList<>();
3      int b = this.q.bitLength();
4
5      for (int i = 0; i < b; i++) {
6          BigInteger two = new BigInteger("2");
7          BigInteger d = two.pow(i) ;
8          Element gen = this.gg.duplicate().
               getImmutable();
9          Element new_e = gen.pow(d).getImmutable();
10         Y.add(new_e);
11     }
12
13     return Y;
14 }
```

2. Pel que fa al vehicle, tindrà dues funcionalitats: la de generar la signatura online en cas de ser el vehicle emisor i la de verificar la signatura en cas de ser el vehicle receptor:

   (a) **Signatura Online**. En aquest cas la funció genera els tres components necessaris per posteriorment generar la signatura: Y,R i z:

```
1  public Pair onlineSign(List<Element> Yp) {
2      String m = "Hello message";
3      BigInteger q = this.rsu.getRTA().getQValue();
4      Pairing e = PairingFactory.getPairing("params.
           params");
5      BigInteger y = e.getZr().newRandomElement().
           toBigInteger();
6      Element Y = e.getZr().newOneElement();
7
8      BitSet bitset = BitSet.valueOf(y.toByteArray());
9      System.out.println(bitset);
10     int b = bitset.length();
11
12     BigInteger num = y;
13     int i = 0;
14
15     while (num.compareTo(new BigInteger("2")) >= 0) {
16         BigInteger quo = num.divide(new BigInteger("2
               "));
17         BigInteger rem = num.remainder(new BigInteger
               ("2"));
18         num = quo;
19         if (rem.compareTo(BigInteger.ONE) == 0) {
20             Y.mul(Yp.get(i));
21          }
22          i++;
23     }
24     if (num.compareTo(BigInteger.ONE) == 0) {
25         Y.mul(Yp.get(i));
26     }
27
28     System.out.println("Y = g^y: " + Y + " = " + this
           .rsu.getRTA().getGValue().pow(y));
29
30     BigInteger h = hashFunction(Y, (Element) this.
           IboosSecretKey.getFirst(), m);
```

```
31      BigInteger s = (BigInteger) this.IboosSecretKey.
            getSecond();
32      BigInteger var = h.multiply(s);
33      BigInteger z = y.add(var).mod(q.subtract(
            BigInteger.ONE));
34
35      return new Pair(Y, z);
36  }
```

Per a la signatura digital, per a comprovar quins bits de y prenen com a valor 1, es va haver de passar y al seu valor en binari i, per fer-ho, es va utilitzar el mètode tradicional de divisions per 2, com es pot veure en la funció anterior en les línies de la 15 a la 23.

(b) **Verificació**. De la signatura online que hem aconseguit del pas anterior (Y, R, z) i amb el missatge m i la identitat ID, comprovem si la signatura és correcta de la següent forma:

```
1  public boolean verifySign(Element Y, Element R,
       BigInteger z, String ID) {
2       BigInteger h = hashFunction(Y, R, "Hello message"
            );
3       BigInteger h2 = this.rsu.getRTA().IBOOSHash(R, ID
            );
4
5       BigInteger var1 = h.multiply(h2);
6       BigInteger pow = this.rsu.getRTA().getXValue().
            pow(var1).toBigInteger();
7       BigInteger mul = Y.mul(R.pow(h)).toBigInteger();
8       BigInteger q = this.rsu.getRTA().getQValue();
9       BigInteger tot = mul.multiply(pow).mod(q);
10      BigInteger first = this.rsu.getRTA().getGValue().
            pow(z).toBigInteger();
11      System.out.println(first + " = " + tot);
12      return true;
13  }
```

5.2.4 Funcionament del sistema IBS/IBOOS proposat

En aquesta secció mostrarem què passa quan executem el nostre codi, és a dir, què es mostra per pantalla en cada pas dels algoritmes i com comprovem que el que generem realment és correcte, aplicant les comprovacions que el paper ens proporciona.

1. Quan el programa comença desde 0, no tenim cap RTA ni cap RSU registrat enllaçat al RTA. Això és el primer que fem:

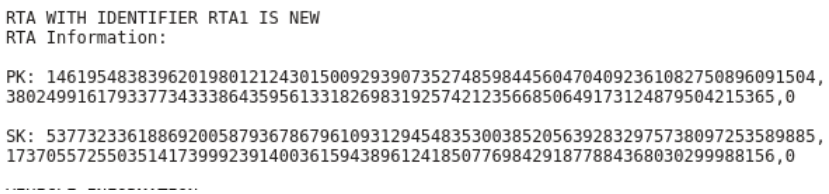

Figura 5.1: Registre d'un nou RTA

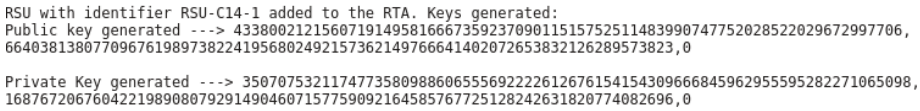

Figura 5.2: Registre d'un nou RSU enllaçat al RSU anterior

Un cop registrats, en la següent execució ens surt un missatge informant-nos de que ja han estat registrats i, per tant, no cal generat noves claus sinó que recuperem les generades en la primera execució.

2. Aleshores el que fem és registrar 3 vehicles en aquest cas per al testing. A continuació una imatge de la sortida produïda pel registre d'un d'aquests tres vehicles. Cal recordar que generem dos parells diferents de claus, un parell utilitzat per a l'esquema IBS i un altre parell utilitzat per al IBOOS:

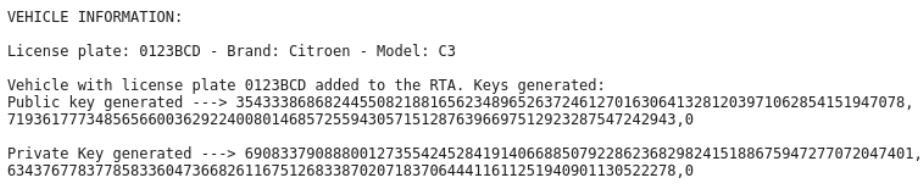

Figura 5.3: Registre d'un nou vehicle.Claus per a l'encriptació

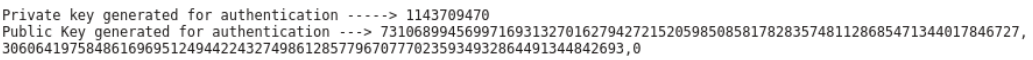

Figura 5.4: Registre d'un nou vehicle. Claus per a l'autenticació

3. Aquestes dades es guarden en la corresponent base de dades:

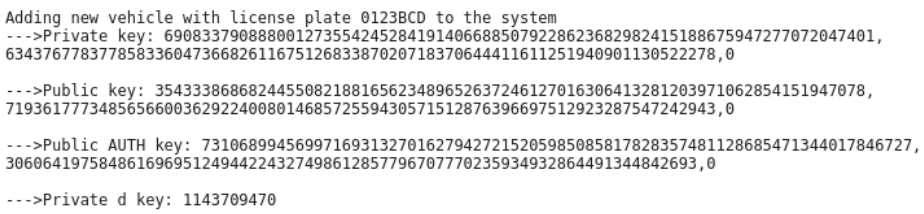

Figura 5.5: Insersió de totes les claus del vehicle en la base de dades

4. En l'algoritme de verificació de la signatura en l'esquema IBS, es comprova que $e(\sigma, P) = e(H, Q)$. Aquesta verificació també es mostra per pantalla per afirmar o no que es compleix, però primer printem el valor de sigma:

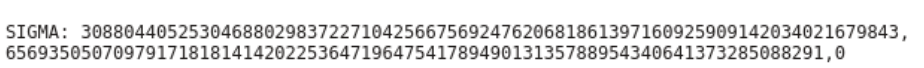

Figura 5.6: Valor de l'atribut σ calculat

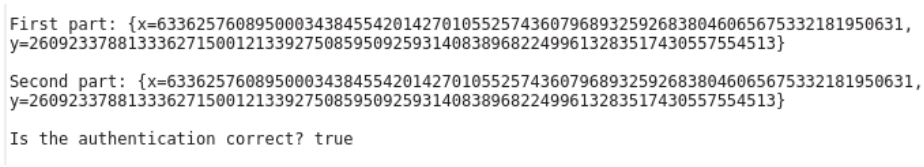

Figura 5.7: Verificació de la igualtat en la signatura de l'esquema IBS

5. Pel que fa a la comunicació entre vehicles utilitzant l'esquema IBOOS, el primer que es mostra per pantalla és el valor de X calculat fent $X = g^x$ on x és la clau mestra secreta i g es un generador de $\mathbb{G}$. A

continuació, en el segon component, en el de l'extracció, per comprovar que la clau secreta és correcta comprovem que $g^s = RX^{H(R,ID)}$. Per recordar el significat de totes les variables, es pot consultar la secció 4.5. La sortida és la següent:

```
IBOOS PART STARTING:
X = g^x = 750181764182670117805428006127531469l3

First part of equality: 507116695187346892873838131126827941 81
Second part of equality: 507116695187346892873838131126827941 81
```

Figura 5.8: Verificació de la igualtat $g^s = RX^{H(R,ID)}$ en l'esquema IBOOS

6. A continuació, i després de generar la signatura offline, generem la signatura online. Després de la generació, i de ser enviada la signatura, el receptor passa a la verificació. Tenim dues comprovacions a realitzar per tal de comprovar la validesa de la signatura. Una és comprovar que $g^z = YR^hX^{hH(R,ID)}$ i l'altra és comprovar que es compleix que $Y = g^y$. Aquestes dues comprovacions sí que es mostren per pantalla:

```
1550956679005443260190487599511517431 2 = 1550956679005443260190487599511517431 2
```

Figura 5.9: Verificació de la igualtat $g^z = YR^hX^{hH(R,ID)}$

```
Y = g^y: 465990331481675623928174383211139580 77 = 465990331481675623928174383211139580 77
```

Figura 5.10: Verificació de la igualtat $Y = g^y$

Capítol 6

Conclusions i futures línies de treball

En aquest treball s'ha realitzat primer, un estudi més teòric sobre conceptes de corbes el·líptiques i aparellaments (o pairings) recollint ja informació utilitzada en l'anterior Treball final de Grau. A més a més, s'ha introduït de manera bastant general el concepte de xarxes vehiculars (o VANETs) incidint més en possibles problemes que plantegen i possibles atacs.

Per poder veure d'una manera menys teòrica com funcionen les comunicacions entre els nodes que pertanyen a aquest tipus de xarxes, es va decidir implementar l'esquema explicat en [10] i [15] , el qual em va semblar molt interessant per poder continuar treballant amb la criptografia basada en la identitat, en el que s'utilitza la combinació de l'esquema IBS (Signatura basada en la Identitat) i l'esquema IBOOS (Signatura Online/Offline basada en la Identitat) per a la comunicació entre vehicle i components externs a la carretera i comuncació entre vehicles.

En aquest cas particular d'implementació, no es poden extreure resultats d'eficiència, ja que la finalitat no era comprovar l'eficiència del programa si no estudiar l'esquema i la seva funcionalitat. Així doncs les conclusions que podem extreure són a nivell teòric.

La criptografia cada dia va assolint un paper més important en la societat, ja que tots els usuaris el que més demanem és garantia de que les nostres dades estaran segures en tot moment. Així que no els estudis relacionats amb

la seguretat cada vegada són més importants i se'n realitzen més. Durant el treball he anat esmentant alguns possibles problemes que podien presentar els esquemes que havien sigut extrets d'articles ja publicats, com poden ser el cost elevat de l'algoritme de búsqueda dins de la llista de revocacions o el possible coll d'ampolla que es podria produir si molts vehicles es volguessin autenticar a la vegada amb l'RTA. Alguns d'aquests problemes podrien ser corregits o millorats, però no és el cas en aquest treball, sinó que podria considerar-se com a futura línia de treball per millorar l'eficiència de l'esquema.

Com a futures línies de treball també es podria seguir treballant amb criptografia basada en la identitat perque com ja he pogut demostrar tant en el treball final de grau com en aquest treball, és un tipus de criptografia que pot ser aplicada a moltes àrees diferents i que de ben segur pot aplicar-se a algunes que encara no han estat estudiades en profunditat.

Bibliografia

[1] M.B.Barbosa, Identity Based Cryptography From Bilinear Pairings, (2005) https://repositorium.sdum.uminho.pt/bitstream/1822/3813/1/report.pdf

[2] P. Barreto, M. Naehrig: Pairing-friendly elliptic curves of prime order. In: Preneel, B., Tavares, S. (eds.) SAC 2005. LNCS, vol. 3897. Springer, Heidelberg, pp. 319–331 (2006)

[3] I. Blake, G. Seroussi, N. Smart: Elliptic curves in cryptography, Cambridge University Press, (1999)

[4] D. Boneh, H. Shacham, B. Lynn : Short Signatures from the Weil Pairing. In: Boyd, C. (ed.) ASIACRYPT 2001. LNCS, vol. 2248, Springer Heidelberg , pp 514–532 (2001)

[5] D. Boneh and M. Franklin. Identity-based Encryption from the Weil Pairing. In CRYPTO 2001, volume 2139 of Lecture Notes in Computer Science. Springer Verlag, pp 213-229 (2001)

[6] D. Boneh, X. Boyen, Efficient selective-ID secure identity-based encryption without random oracles, in: Advances in Cryptology—Eurocrypt 2004, in: LNCS, vol. 3027, Springer-Verlag, pp. 223–238 (2004)

[7] R. Crandall, C. Pomerance. Prime Numbers, a Computational perspective (Second Edition), pp. 348-371 (2000)

[8] D. Freeman, Constructing pairing-friendly elliptic curves with embedding degree 10, in Algorithmic Number Theory Symposium—ANTS-VII. Lecture Notes in Computer Science, vol. 4076, Springer Berlin, pp. 452–465 (2006)

[9] S. Galbraith, Pairings, in Advances in elliptic curve cryptography, I. F. Blake, G. Seroussi, N. P. Smart, eds., London Math. Soc. Lect. Note Series 317, Cambridge Univ. Press, Cambridge, pp 211, (2005)

[10] Lu, Huang & Li, Jie & Guizani, Mohsen. A novel ID-based authentication framework with adaptive privacy preservation for VANETs. (2012)

[11] B. Lynn. On the implementation of pairing-based cryptosystems. Ph.D. thesis, Stanford, (2008)

[12] M. N. Mejri J. Ben-Othman M. Hamdi "Survey on VANET security challenges and possible cryptographic solutions" Veh. Commun. vol. 1 no. 2 ,pp. 53-66 (2014)

[13] V. S. Miller. Short Programs for functions on Curves. Unpublished manuscript, (1986)

[14] J. Miret, J. Pujolàs, J. Valera, MOOC Criptografia con emparejamientos, Crypt4you (2017) http://www.criptored.upm.es/crypt4you/temas/ECC/leccion3/leccion3.html

[15] R. Nasreen Salma, N.Alangudi Balaji, Dr.R.Sukumar, " A Framework for Authentication in Vehicular Ad-hoc Network using Identity based approach", IOSR Journal of Engineering (IOSRJEN) vol. 3, no.7, pp. 15-19 (2013)

[16]] Reńe Schoof, Elliptic curve over finite fields and the computation of square roots mod p, Mathematics of Computation 44, no. 170, pp. 483–495 (1985)

[17] R. Sakai, M. Kasahara, ID based cryptosystems with pairing on elliptic curve, Cryptology ePrint Archive, Report 2003/054, (2003)

[18] A. Shamir, "Identity-based cryptosystems and signature schemes", Proc. Crypto' 84, pp. 47–53